(12) UK Patent Application (19) GB (11) 2 217 155 (13) A

(43) Date of A publication 18.10.1989

(21) Application No 8904078.6

(22) Date of filing 22.02.1989

(30) Priority data
(31) 63079539 (32) 31.03.1988 (33) JP
63267940 24.10.1988
63267941 24.10.1988
63267942 24.10.1988

(71) Applicant
Yokogawa Electric Corporation

(Incorporated in Japan)

9-32 Nakacho 2-chome, Musashino-shi, Tokyo 180, Japan

(72) Inventor
Kenichi Inoue

(74) Agent and/or Address for Service
Hughes Clark & Co
63 Lincoln's Inn Fields, London, WC2A 3JU, United Kingdom

(51) INT CL⁴
G06F 3/153

(52) UK CL (Edition J)
H4T TBAX
G4A AMX
U1S S2284

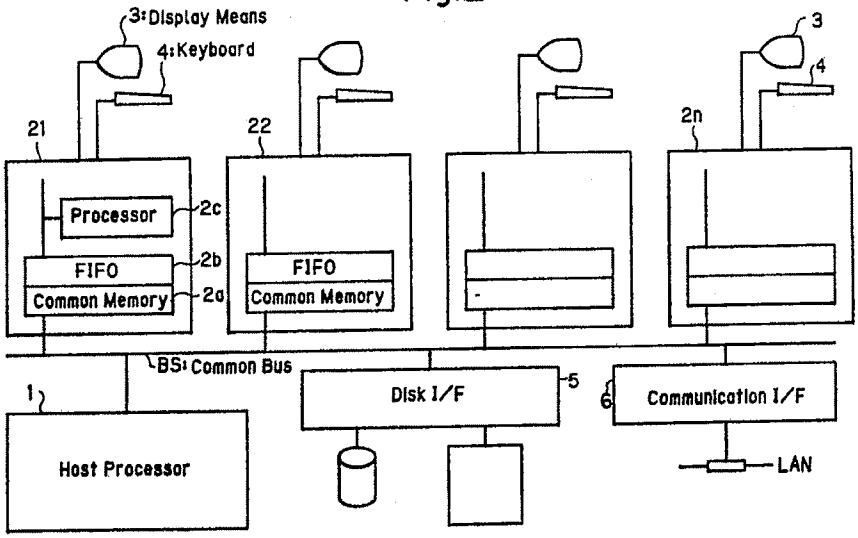
(56) Documents cited
None

(58) Field of search
UK CL (Edition J) G4A AKB1 AMT1, H4T TBAD TBAG TBAX
INT CL⁴ G06F

(54) Graphic display system

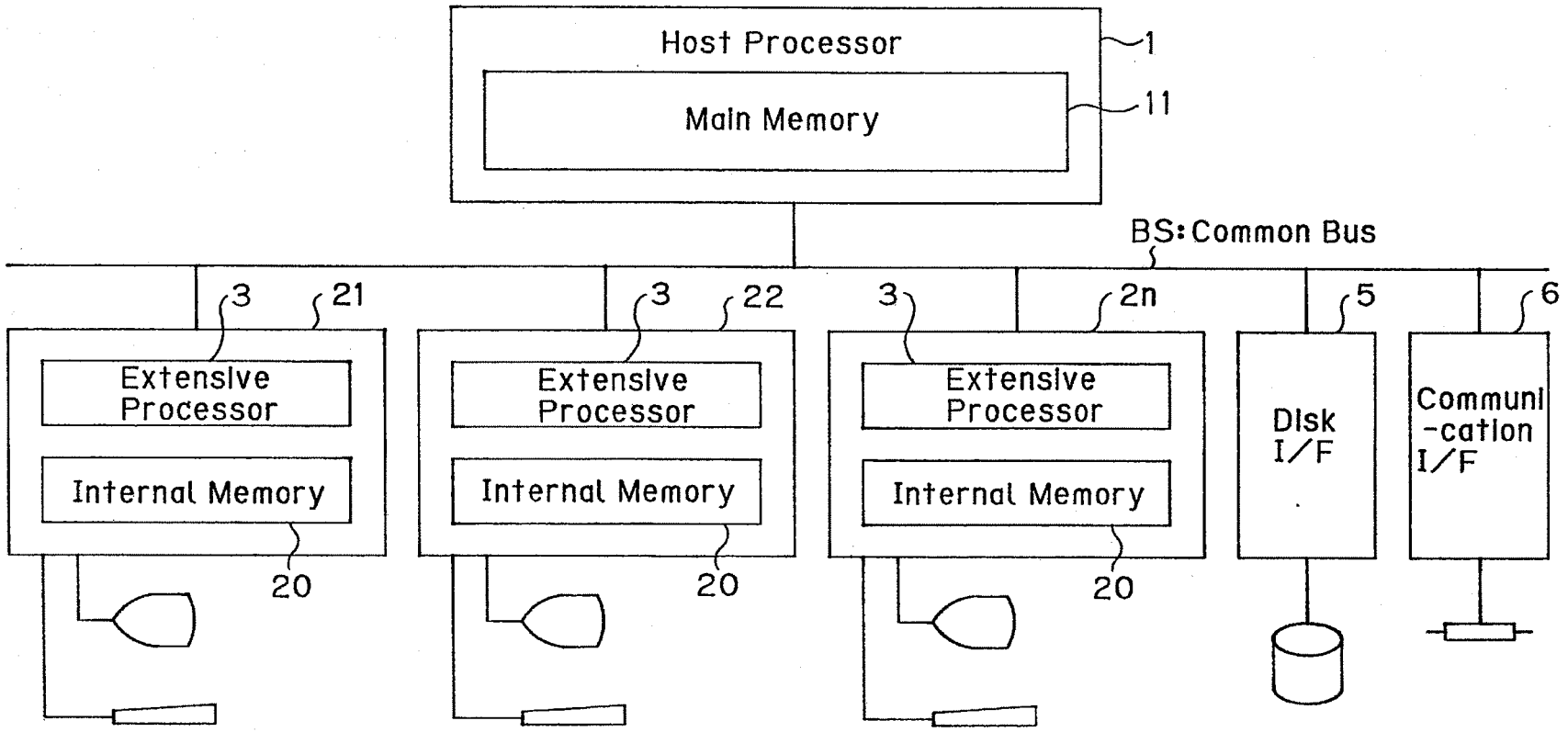
(57) A system having a plurality of graphic display units (2_1 to 2_n) connected to the host processor (1) through a common bus BS provides useful quick response characteristics as a man-machine interface for process control. A 2-port common memory (2a) one port of which is connected to the common bus and the other to an internal bus is provided in each graphic display unit and a part of this common memory in the form of a First-In-First-Out buffer can be used for transmission of the high level command/data with the host processor (1), under the control of PUSH (host processor to common memory) and POP (common memory to internal bus) FIFO pointers (P1, P2).

Fig.2



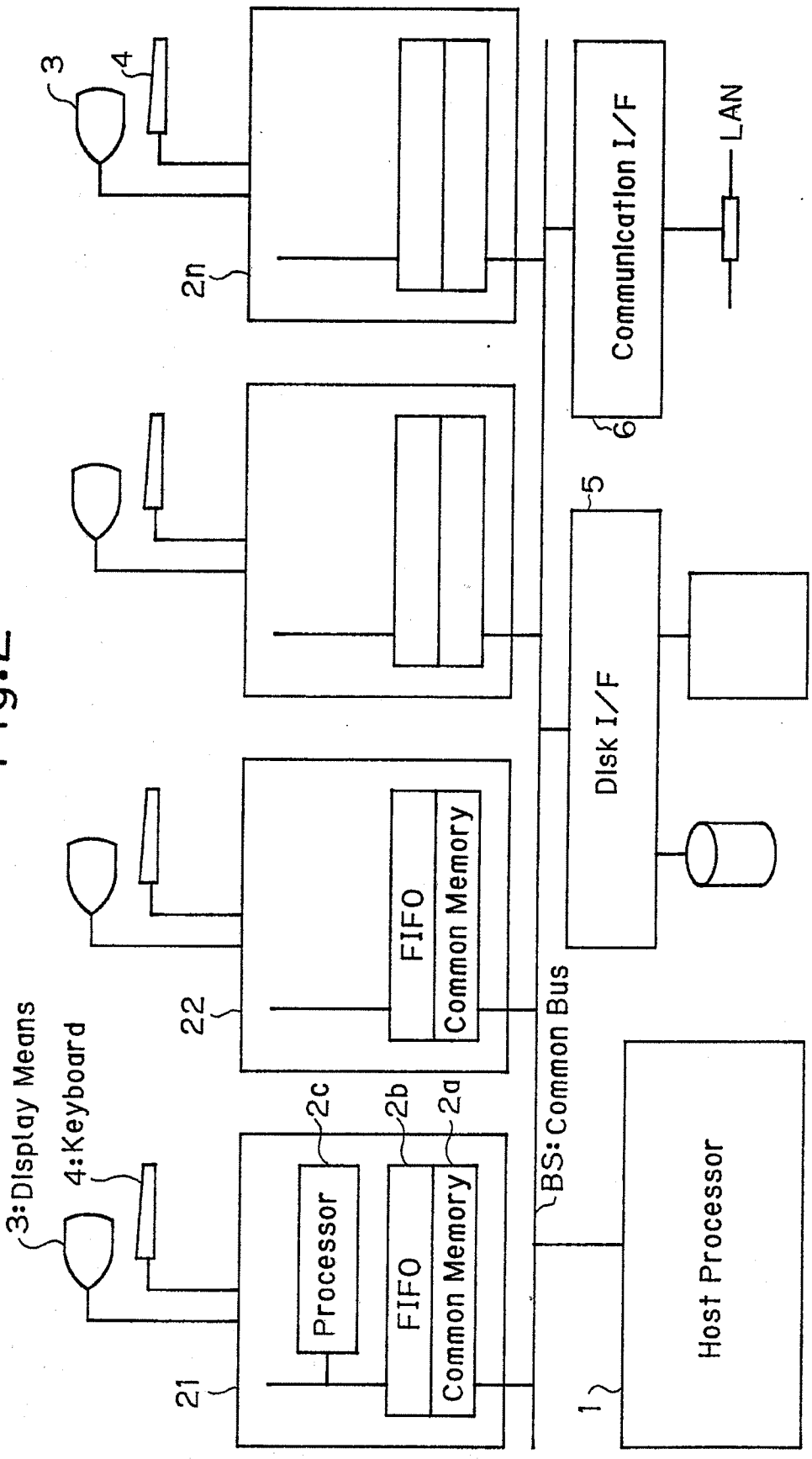
GB 2 217 155 A

Fig.1 (Prior Art)



2217155

Fig.2


3:Display Means
4:Keyboard
3
4
2n
2c
2b
2a
21
22
Processor
FIFO
Common Memory
FIFO
Common Memory
BS: Common Bus
Host Processor
1
Disk I/F
5
Communication I/F
6
LAN

2217155

Fig.3

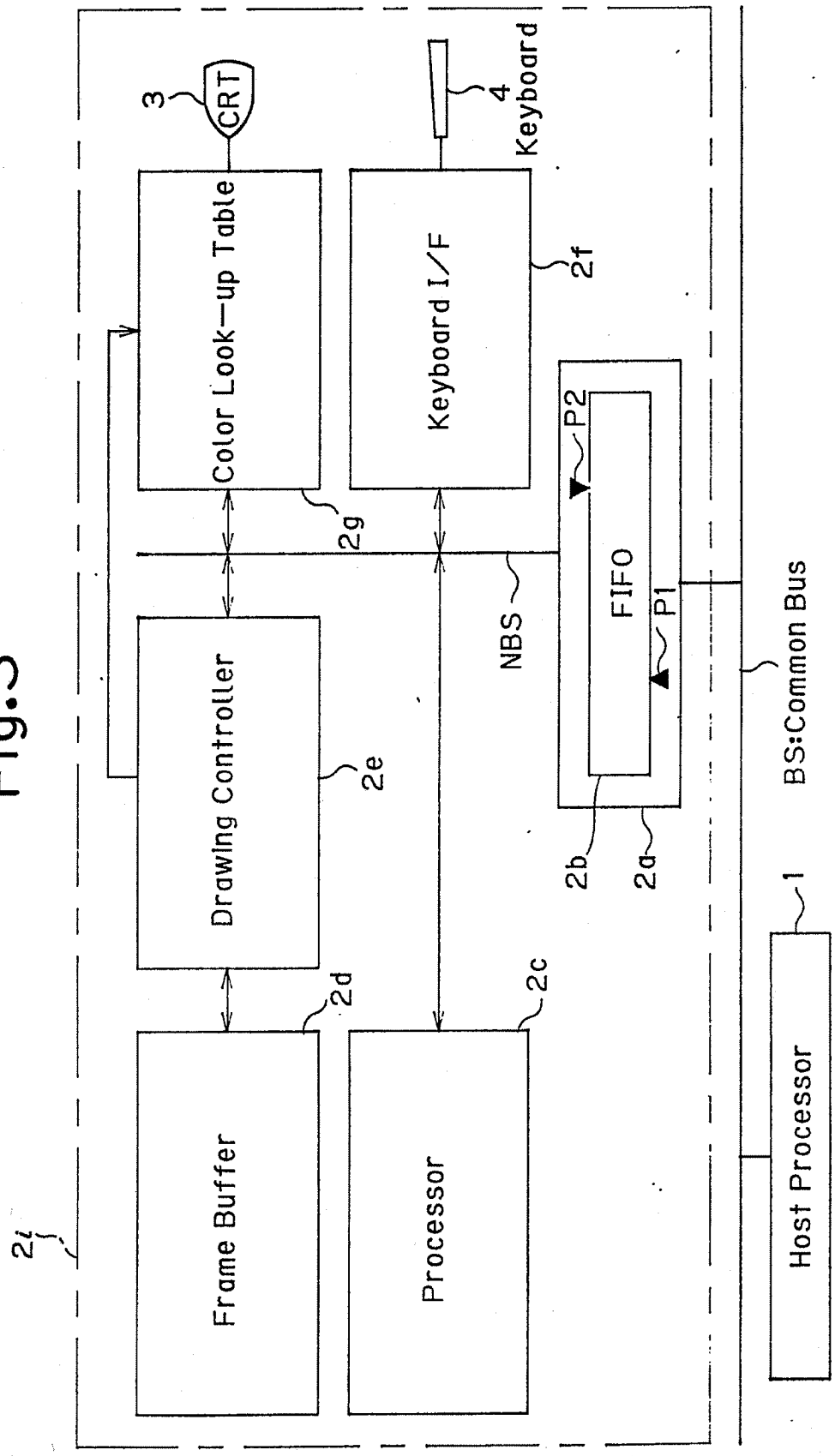


Fig.4

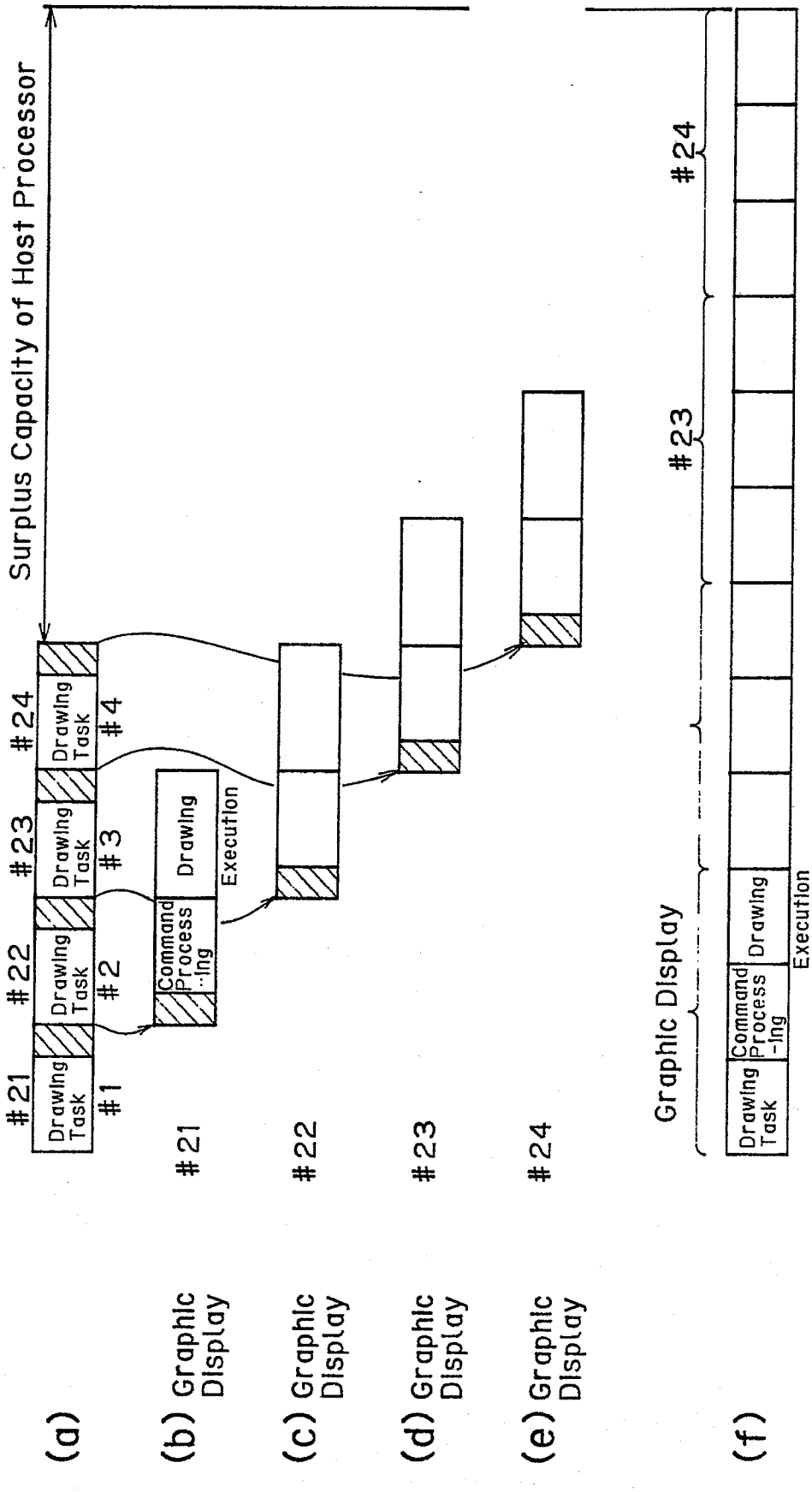

(a)
Surplus Capacity of Host Processor
#21
Drawing Task
#1
#22
Drawing Task
#2
#23
Drawing Task
#3
#24
Drawing Task
#4
(b) Graphic Display
#21
Command Process -ing
Drawing Execution
(c) Graphic Display
#22
(d) Graphic Display
#23
(e) Graphic Display
#24
(f)
Graphic Display
#23
#24
Drawing Task
Command Process -ing
Drawing Execution

Fig.5

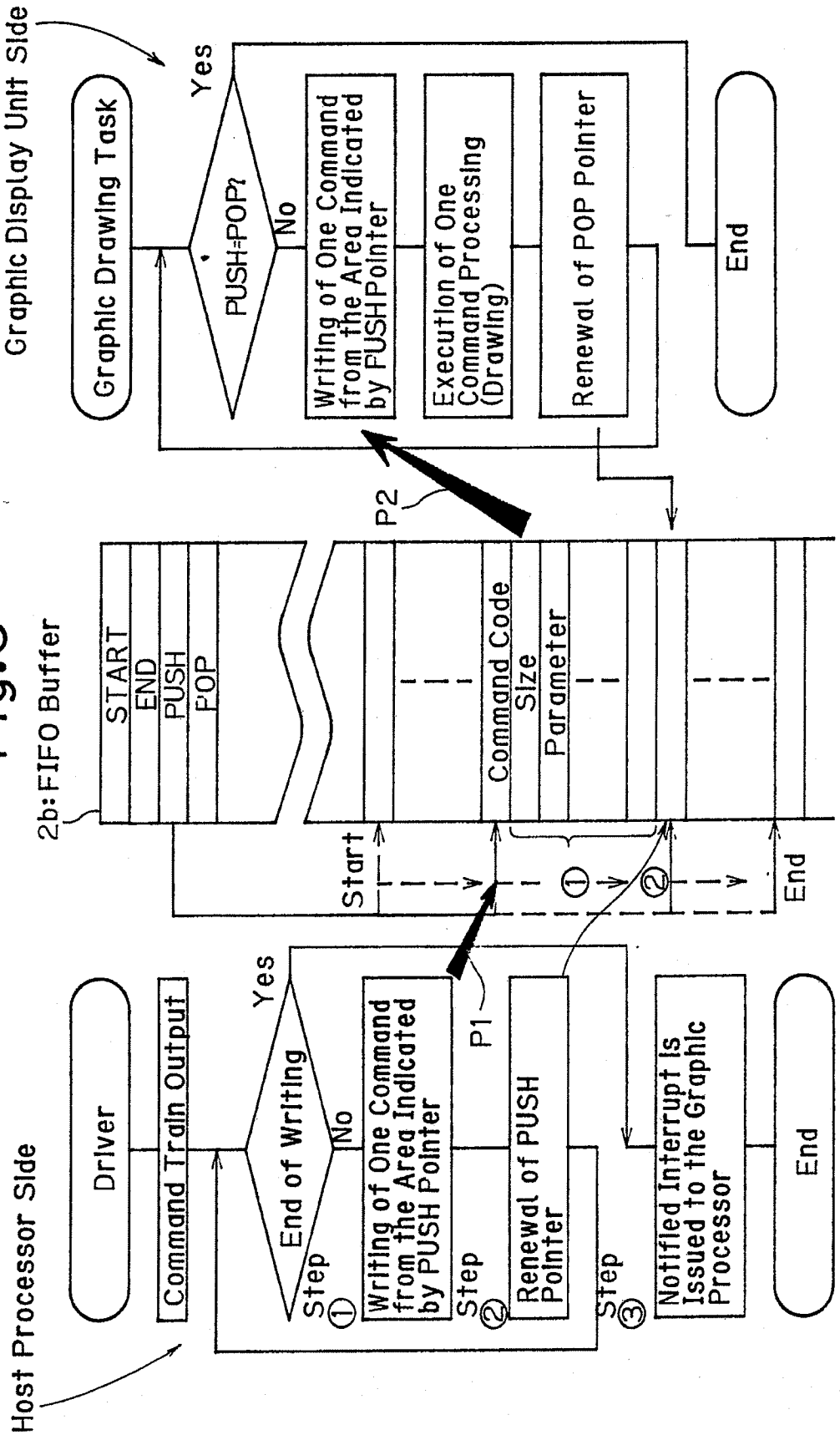

Host Processor Side
Driver
Command Train Output
Step ①
End of Writing
Yes
No
Writing of One Command from the Area Indicated by PUSH Pointer
Step ②
Renewal of PUSH Pointer
Step ③
Notified Interrupt is Issued to the Graphic Processor
End
P1
2b: FIFO Buffer
START
END
PUSH
POP
Start
Command Code
Size
Parameter
End
P2
Graphic Display Unit Side
Graphic Drawing Task
PUSH=POP?
Yes
No
Writing of One Command from the Area Indicated by PUSH Pointer
Execution of One Command Processing (Drawing)
Renewal of POP Pointer
End

Fig.6

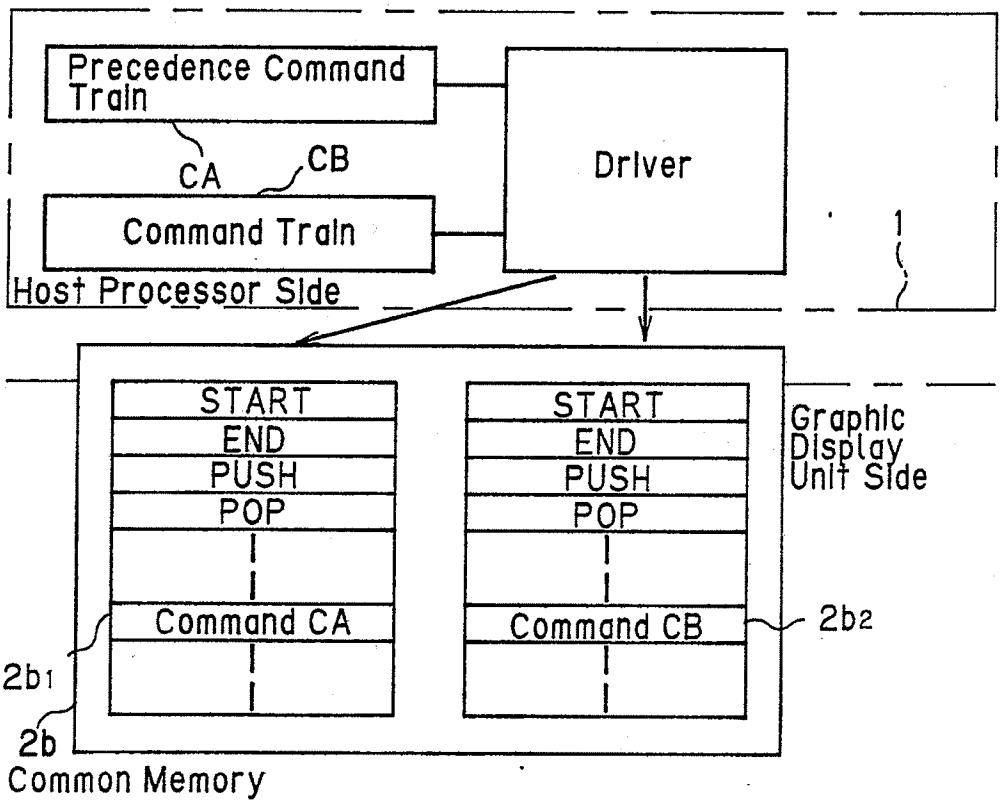

Precedence Command Train
CA
CB
Command Train
Driver
1
Host Processor Side
START
END
PUSH
POP
Command CA
Command CB
Graphic Display Unit Side
2b1
2b2
2b
Common Memory

Fig.7

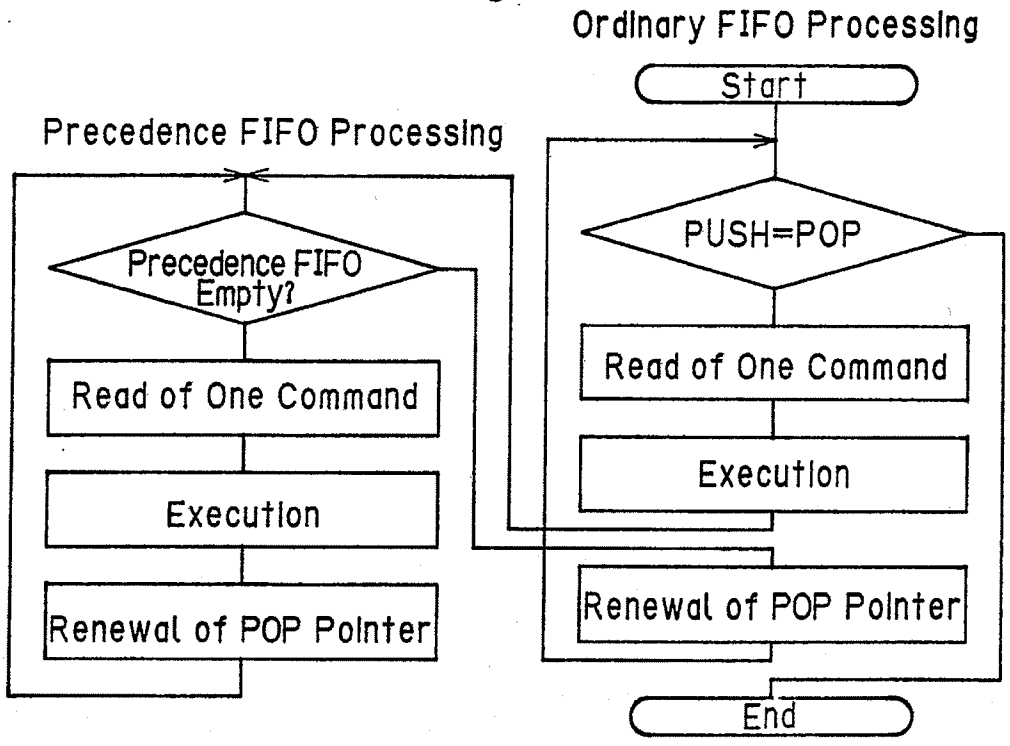

Ordinary FIFO Processing
Start
Precedence FIFO Processing
Precedence FIFO Empty?
PUSH=POP
Read of One Command
Execution
Renewal of POP Pointer
Read of One Command
Execution
Renewal of POP Pointer
End

Fig.8

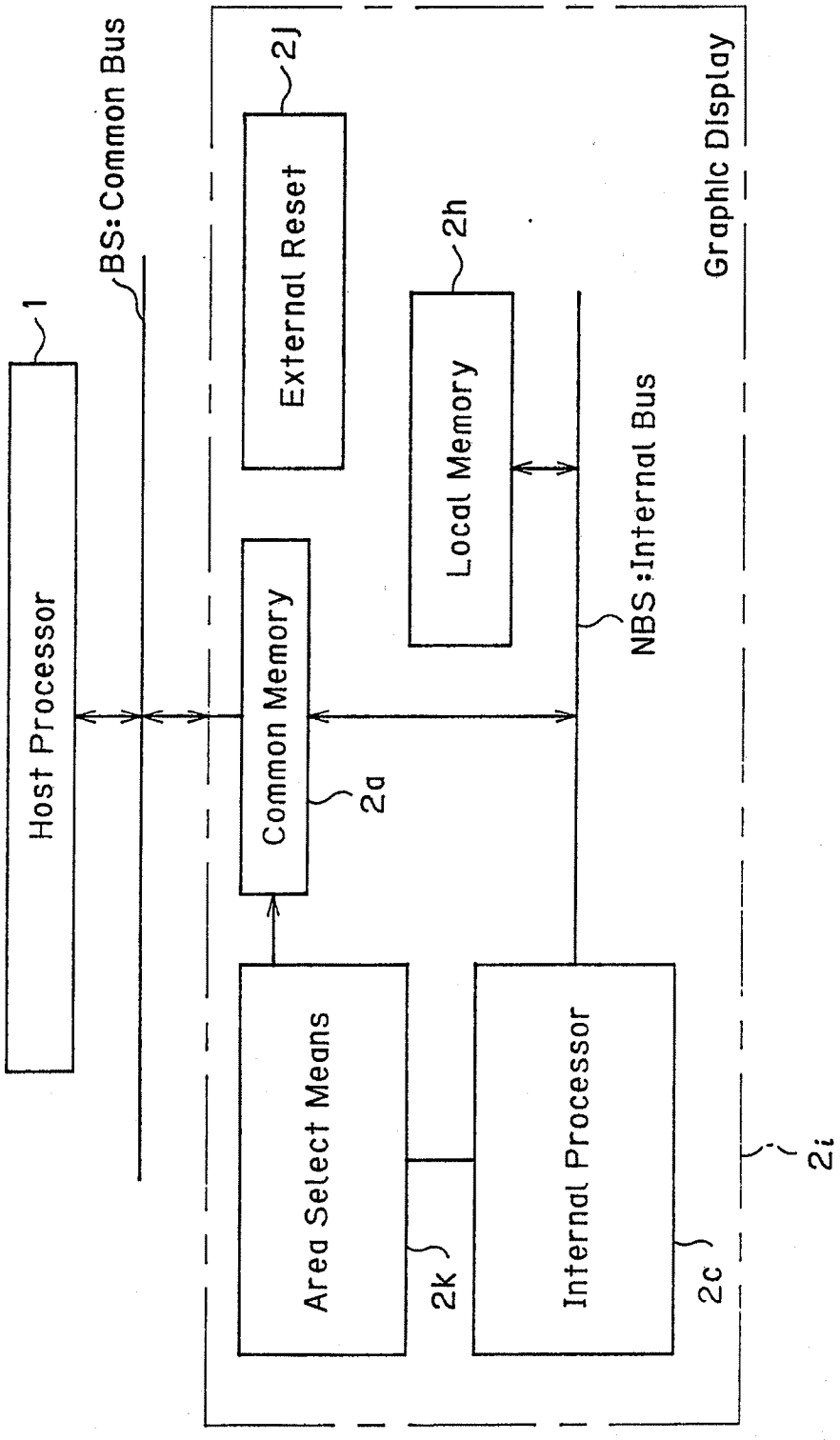

Host Processor
1
BS: Common Bus
External Reset
2J
Local Memory
2h
Common Memory
2a
NBS : Internal Bus
Graphic Display
Area Select Means
2k
Internal Processor
2c
2i

Fig.9

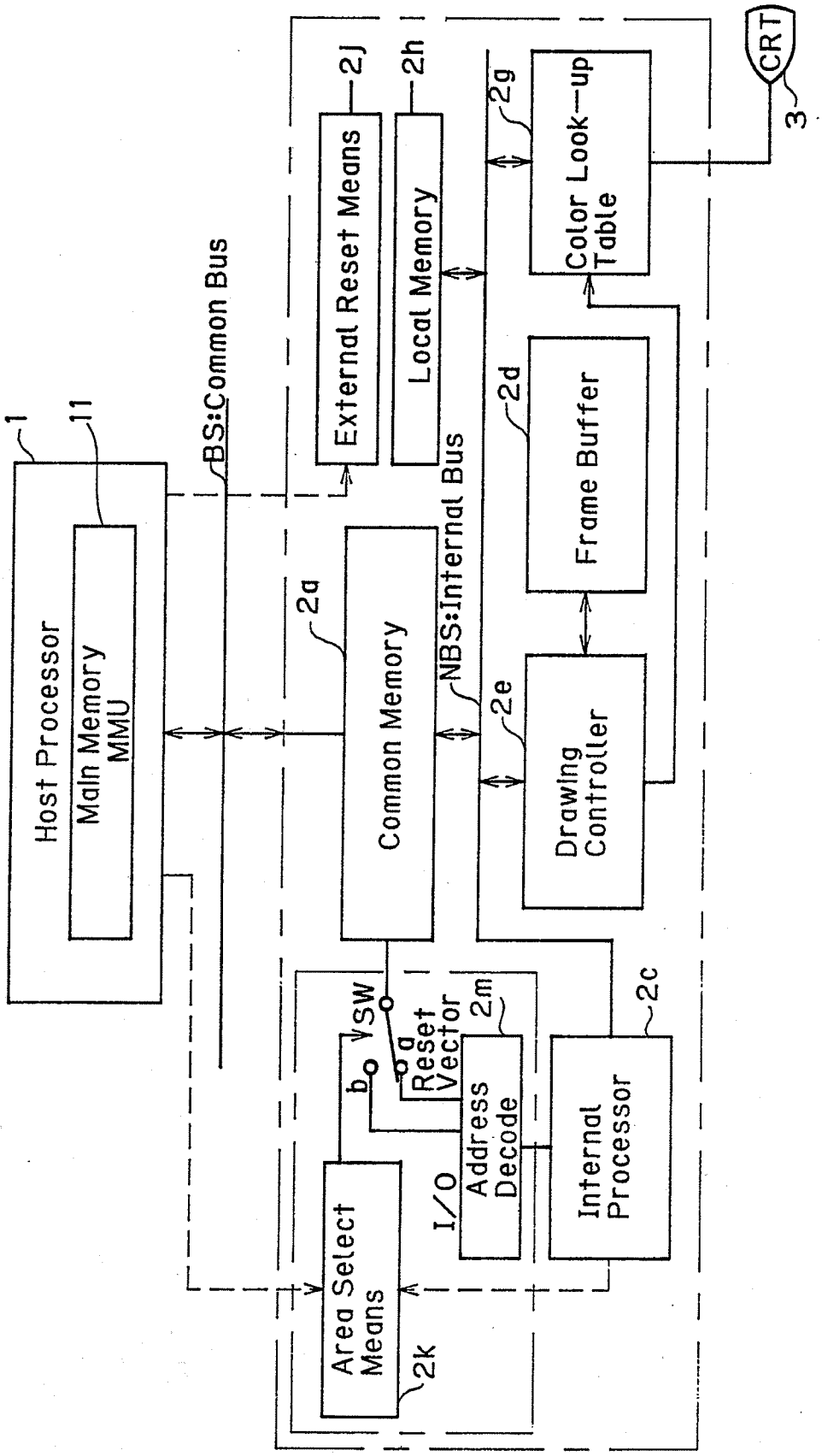

Host Processor
Main Memory MMU
1
11
BS:Common Bus
External Reset Means
2J
Local Memory
2h
Color Look-up Table
2g
CRT
3
2a
Common Memory
NBS:Internal Bus
Frame Buffer
2d
Drawing Controller
2e
SW
b
a
Reset Vector
2m
I/O
Address Decode
Internal Processor
2c
Area Select Means
2k

Fig.10

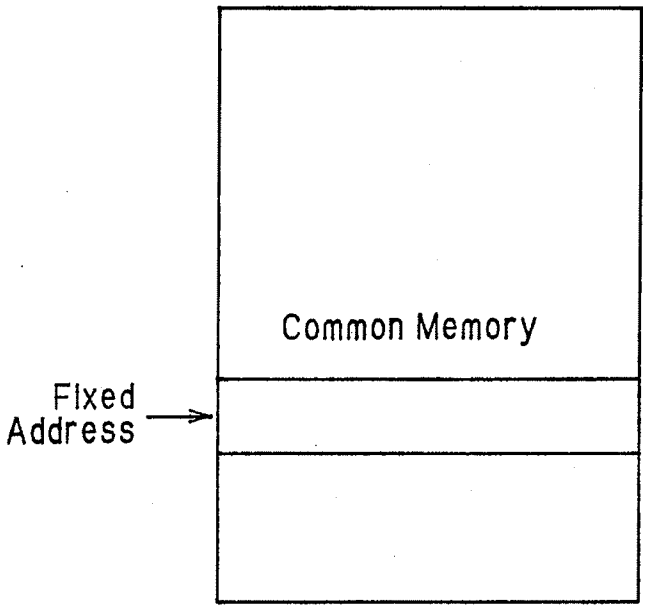


Fig.11(a)

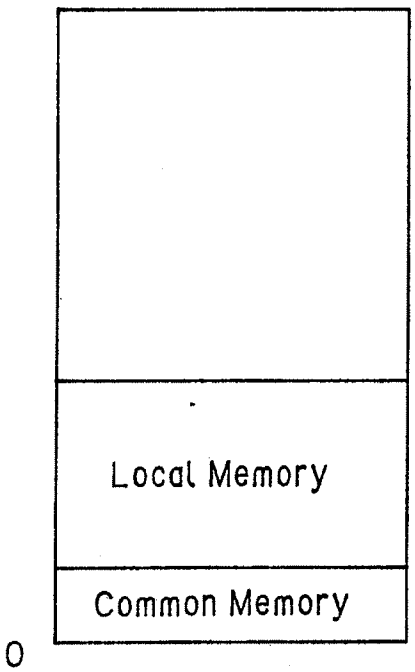


Fig.11(b)

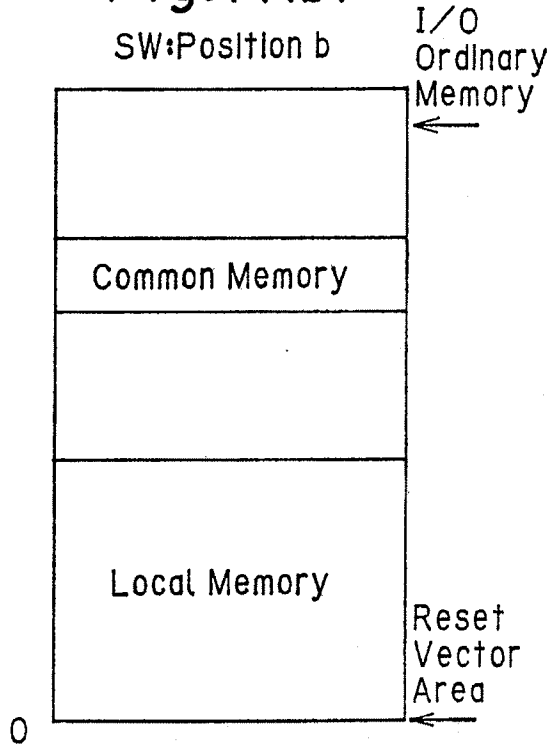


Fig.12

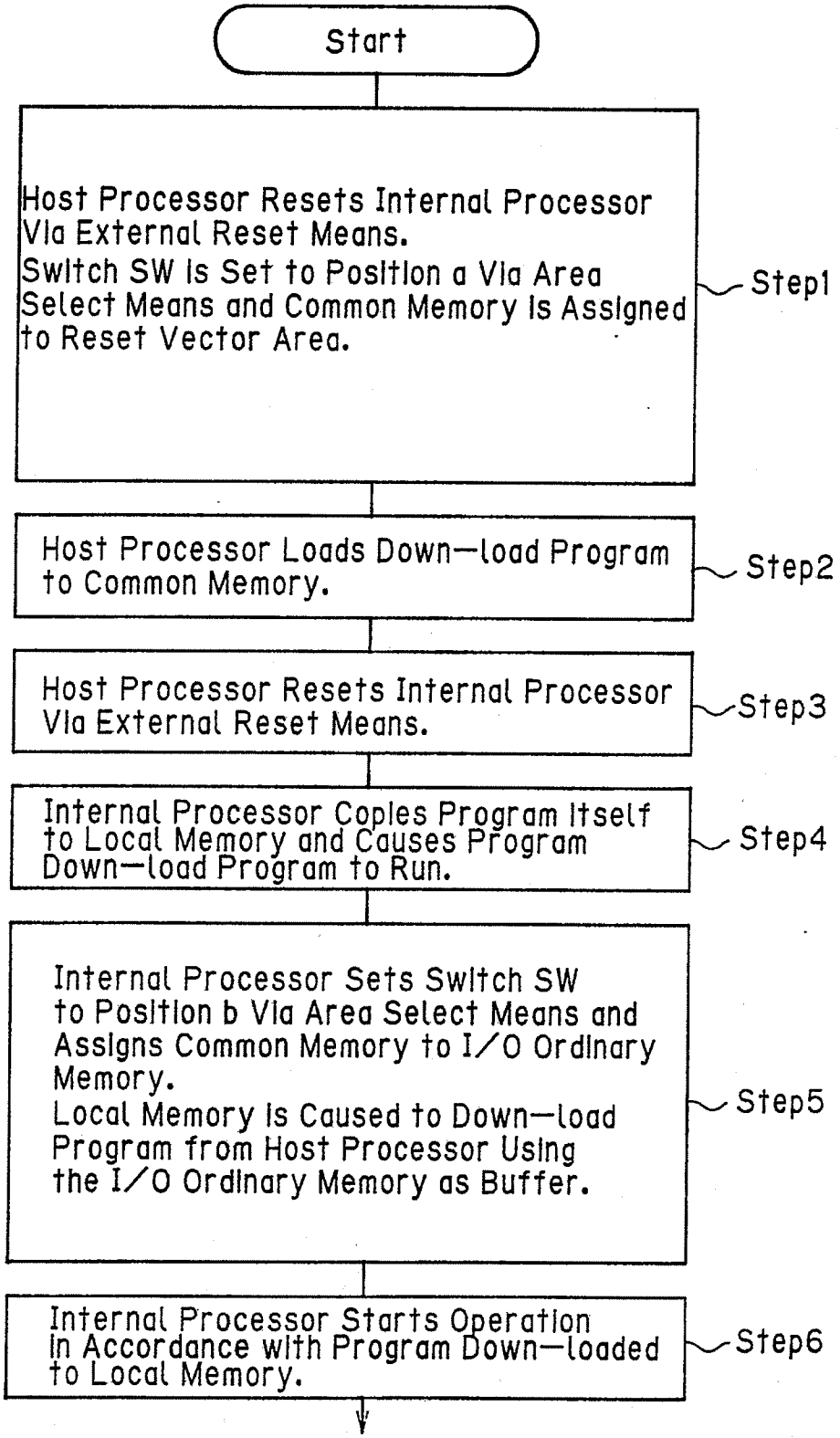
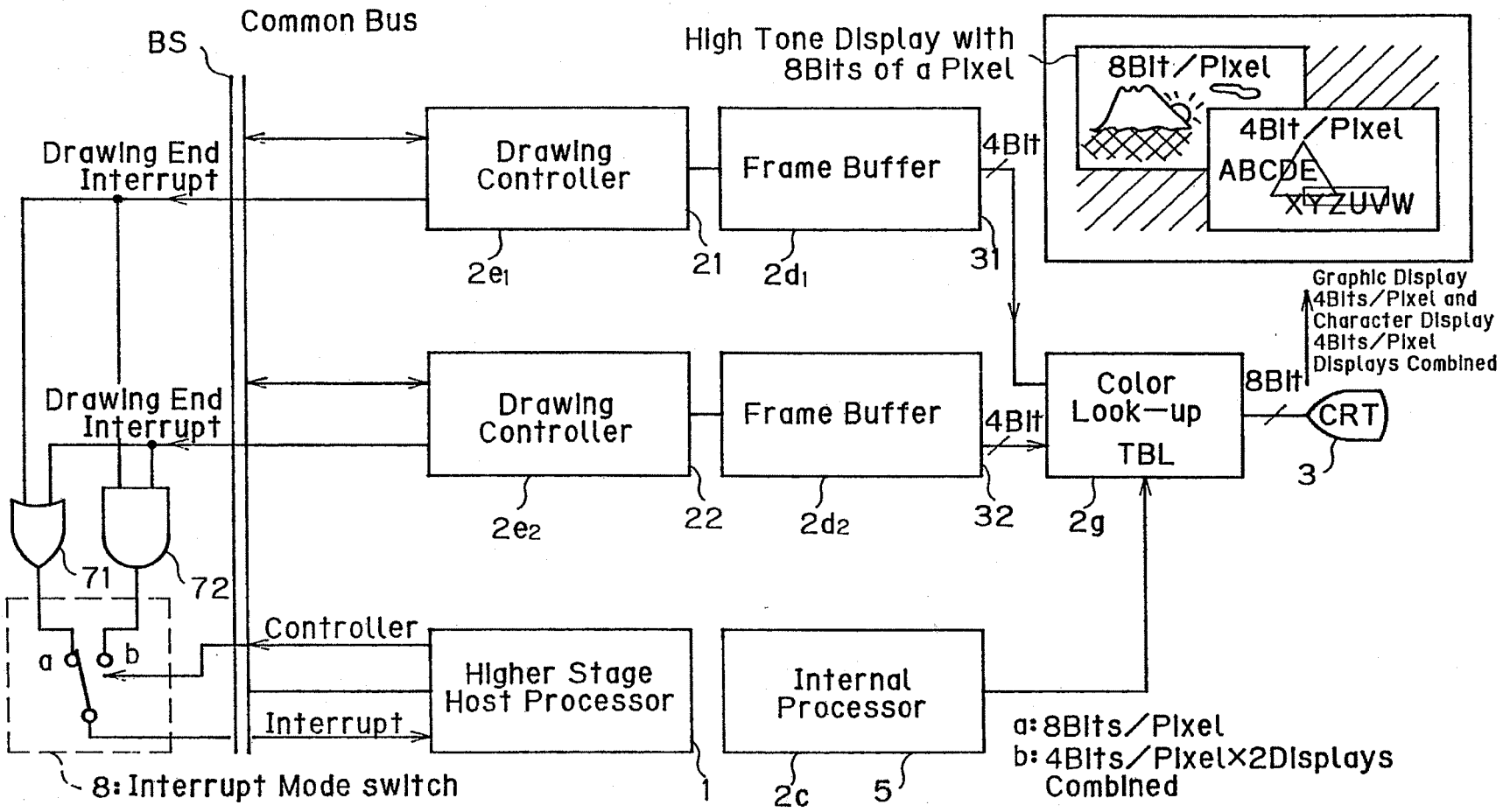

Start
Host Processor Resets Internal Processor Via External Reset Means.
Switch SW is Set to Position a Via Area Select Means and Common Memory is Assigned to Reset Vector Area.
Step1
Host Processor Loads Down—load Program to Common Memory.
Step2
Host Processor Resets Internal Processor Via External Reset Means.
Step3
Internal Processor Copies Program Itself to Local Memory and Causes Program Down—load Program to Run.
Step4
Internal Processor Sets Switch SW to Position b Via Area Select Means and Assigns Common Memory to I/O Ordinary Memory.
Local Memory is Caused to Down—load Program from Host Processor Using the I/O Ordinary Memory as Buffer.
Step5
Internal Processor Starts Operation In Accordance with Program Down—loaded to Local Memory.
Step6

Fig.13



2217155

Fig.14

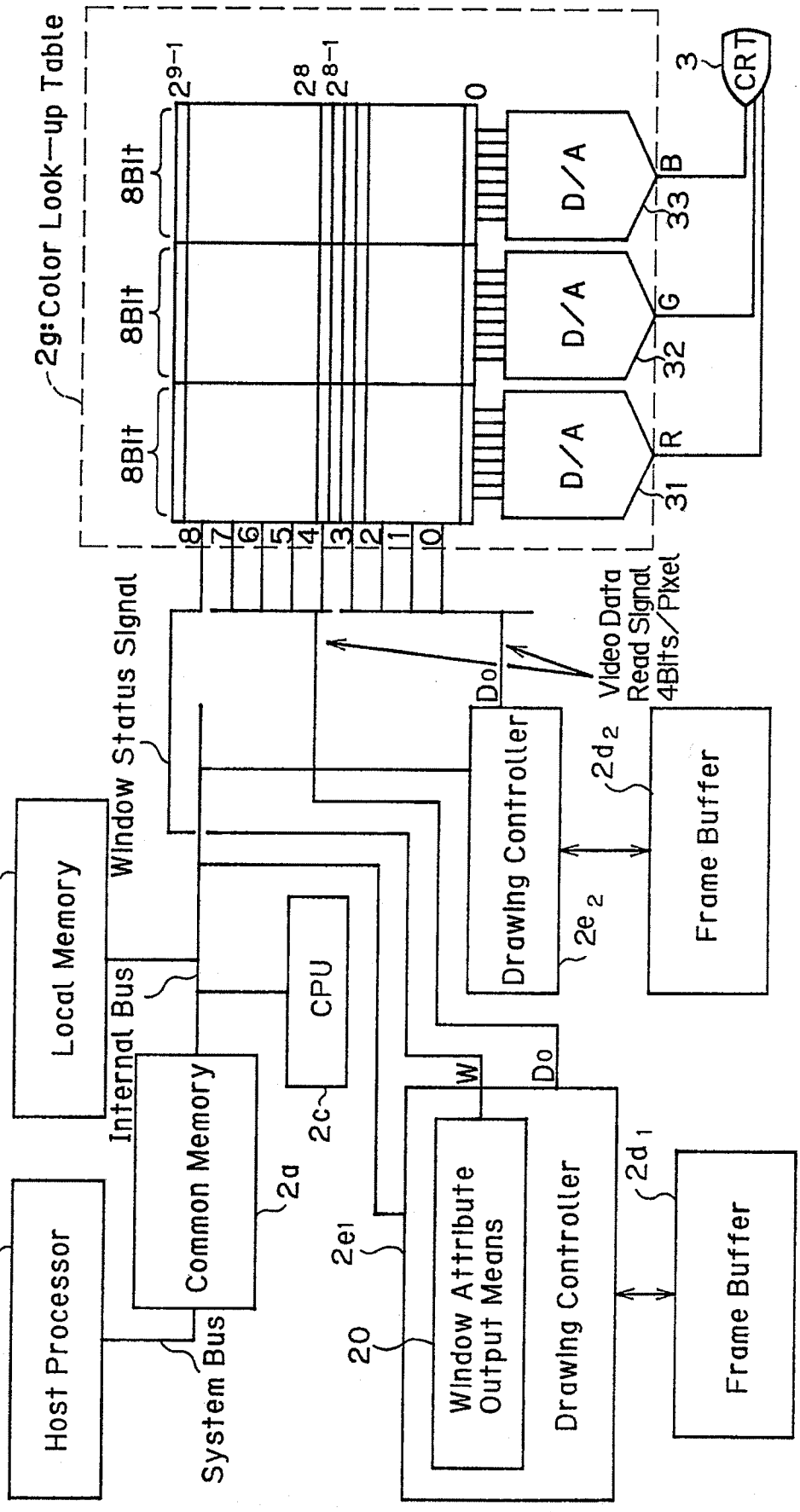

2g:Color Look-up Table
8Bit
8Bit
8Bit
$2^{9}-1$
2^{8}
$2^{8}-1$
0
D/A
D/A
D/A
B
G
R
33
32
31
3
CRT
8
7
6
5
4
3
2
1
0
Do
Video Data Read Signal 4Bits/Pixel
Window Status Signal
2h
Local Memory
Internal Bus
CPU
2c
Drawing Controller
$2d_2$
$2e_2$
Frame Buffer
W
Do
Common Memory
2a
2e1
20
Window Attribute Output Means
Drawing Controller
$2d_1$
Frame Buffer
1
Host Processor
System Bus

Fig.15

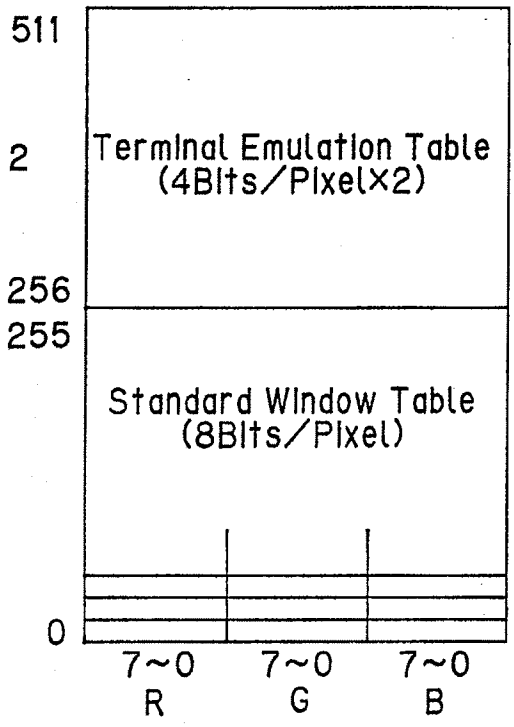


Fig.16

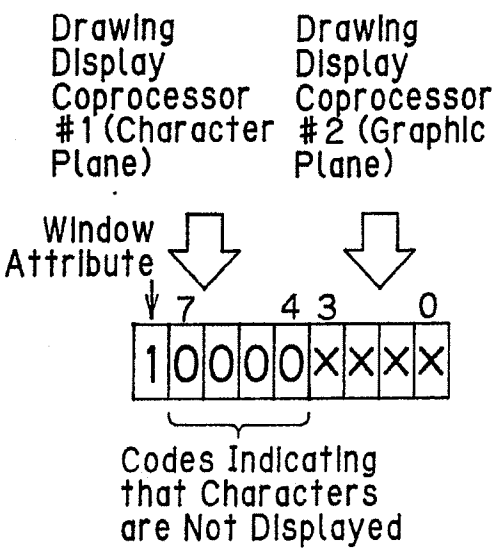


Fig.17(b)

Graphic Plane

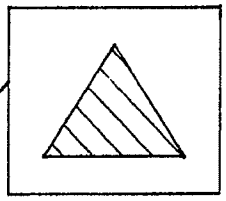


Fig.17(c)

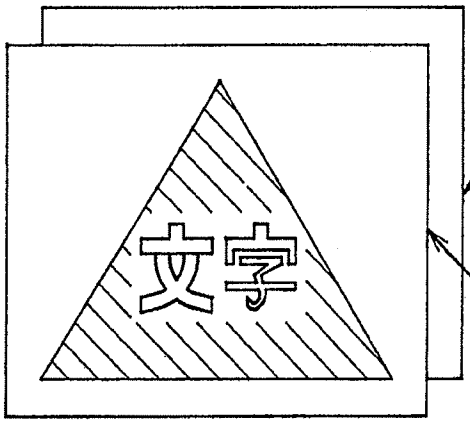
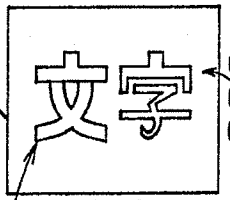
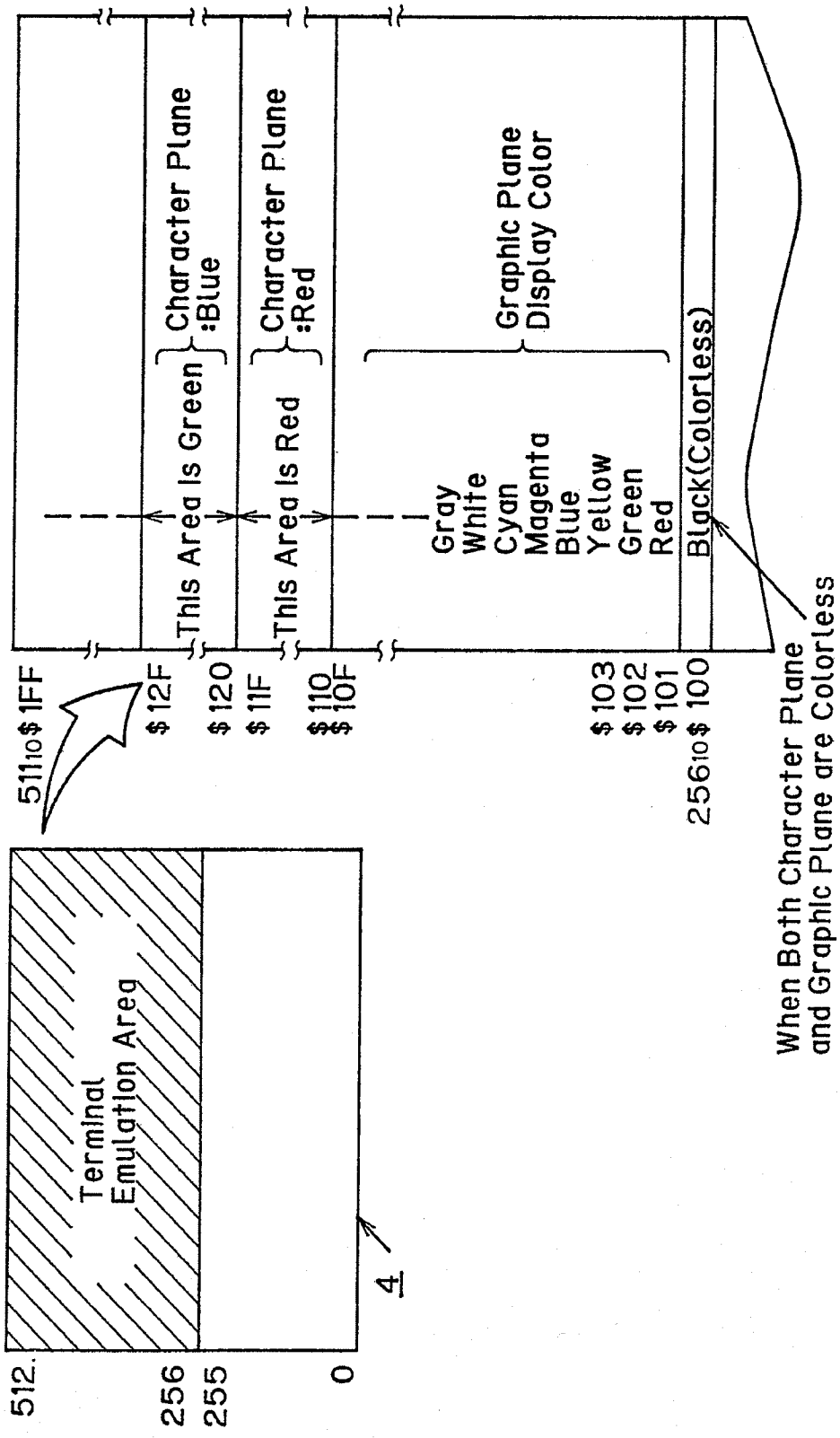


Fig.17(a)



Color Code Character Plane other than 0000

Fig.18


512
256
255
0
Terminal Emulation Area
4
511₁₀ $1FF
$12F
$120
$11F
$110
$10F
$103
$102
$101
256₁₀ $100
This Area Is Green
Character Plane :Blue
This Area Is Red
Character Plane :Red
Gray
White
Cyan
Magenta
Blue
Yellow
Green
Red
Graphic Plane Display Color
Black(Colorless)
When Both Character Plane and Graphic Plane are Colorless

GRAPHIC DISPLAY SYSTEM

The present invention relates to a graphic display unit which can be effectively used for a man-machine interface for process control and particularly for a system with a configuration in which a plurality of graphic display units are connected to a host processor through a common bus in which each graphic display unit receives drawing commands and display commands sent from the host processor and displays in parallel graphics and patterns on a display unit such as a CRT in accordance with applied information.

A graphic display unit to be used as a man-machine interface for process control involves the problem of effectively displaying a variety of information at a high speed on a display means for managing and controlling the plant as a whole.

The present invention has been proposed considering such problem of the prior art and therefore it is a principal object of the present invention to provide a graphic display unit in which the host processor effectively transmits commands to a plurality of display controllers to enhance the efficiency of parallel processing.

It is another object of the present invention to simplify configuration of the graphic display unit and improves the maintenance ability thereof.

It is further object of the present invention to realise an apparatus in which a graphic display and a character display can be superimposed to each other in the one window of a multi-window arrangement, each of such displays can be controlled individually and moreover this window and another standard window are allowed to co-exist for display in the same display screen.

According to the present invention a graphic display unit, in a system connecting a plurality of graphic display units to a host processor via a common bus comprises a 2-port common memory connected to the common bus at the one terminal and to an internal bus at

the other terminal, a part of this common memory being in the form of a FIFO (First-In First-Out) buffer for transmission of high level command/data to the host processor.

In order that the invention may be clearly understood and readily carried into effect examples thereof as well as an example of the prior art will now be described with reference to the accompanying drawings, in which:-

Fig. 1 is a conceptional view of a graphic display unit according to the prior art;

Fig. 2 is a block diagram of an embodiment of a basic configuration according to the present invention;

Fig. 3 is a block diagram showing a portion of Fig. 2 in greater detail;

Fig. 4 is a timechart relating to Figs. 2 and 3;

Fig. 5 is a flowchart pertaining to the graphic display unit of Figs. 2 to 4;

Fig. 6 is a conceptional view another embodiment of the present invention;

Fig. 7 is a flowchart indicating an example of the operations conducted by the embodiment of Fig. 6;

Fig. 8 is a block diagram indicating another basic configuration example of a graphic display unit of the present invention;

Fig. 9 is a block diagram showing a developed version of the display unit of Fig. 8;

Figs. 10, 11a, 11b are explanatory diagrams applicable to Fig. 9;

Fig. 12 is a flowchart indicating an example of the operation of the display unit of Fig. 9;

Fig. 13 is a block diagram indicating the principal portion of another configuration example of a graphic display unit;

Fig. 14 is a block diagram indicating yet another configuration example of a graphic display unit;

Figs. 15, 16, 17a, 17b, 17c, 18 are explanatory diagrams applicable to the display unit of Fig. 14.

In the prior art of Figure 1, 1 designates a host processor connected to a common bus BS; $2_1 \sim 2_n$, graphic display/keyboard controllers for enabling the functioning of a man-machine interface connected to the common bus BS and each including therein an exclusive sub-processor 3. The common bus BS is also connected with an interface for a disk unit 5 and a communication interface 6, in addition to the controllers.

Such existing apparatus is so configurated that if the host processor 1 controls the controllers $2_1 \sim 2_n$, either the controller reads, as the bus master, commands written in the command block of the main memory 11 of the host processor 1 in accordance with an instruction from the host processor 1, or alternatively the host processor 1 tries to obtain a bus right within a controller and transfers a command train by directly writing the commands to the internal memory 20 thereof.

However, in the case of said first configuration, the controller has to become a bus master and therefore the configuration is complicated and moreover delay due to transfer of the bus right increases, resulting in a problem that performance drop can no longer be negligible.

In addition, the second said configuration provides the problem that the processing speed of the subprocessor within the controller is remarkably lowered because the bus right is lost.

Fig. 2 is a block diagram indicating a basic configuration of the present invention. In this figure, the numeral 1 designates a host processor; $2_1 \sim 2_n$, a plurality of graphic display units connected to the host processor 1 through the common bus BS, each providing a display means 3 and a keyboard 4; 5 and 6, interface units such as disk unit and local area network (LAN) connected to the common bus BS.

In each grahic display unit 2_1 2_n, 2a is a common memory having two ports, the one of which is connected to the common bus BS and the other to the internal bus,

which is used in common by the host processor and an internal processor 2c, and 2b is a FIFO buffer provided to a part of the common memory 2a. The internal processor 2c is provided for sharing a part of the processes carried out by the host processor 1 and is connected with the common memory 2a through the internal bus.

Fig. 3 is a block diagram of configuration indiating an embodiment of the graphic display unit. In this figure, the internal configuration of only one graphic display unit is indicated and it is similar to that of the other graphic display units.

Within the common memory 2a, the FIFO buffer 2b is formed according to the software and the write and read operations of the commands and data can be controlled by a PUSH pointer P1 which instructs a write area of data from the host processor 1 and a POP pointer P2 which indicates the data read to an end area of processor 2c.

The other area of this common memory 2a is an ordinary common area which allows entry of program code or is used for transmission of other information.

2d designates frame buffer; 2e, drawing controller; 2f, keyboard interface; 2g, colour look-up table. The drawing controller 2e decodes commands and data read from the FIFO buffer 2b, draws necessary pattern or characters on the frame buffer 2d or reads them in accordance with the decoded commands and data, and executes graphic display on the display means 3 via the colour look-up table 2g.

Operations of the apparatus thus configurated will now be explained hereunder.

Fig. 4 is a time chart indicating the sharing of processes by respective devices. For a plurality of graphic display units 2_1, 2_2, ..., 2_n, the host processor 1 sequentially writes, as shown in (a), high level commands, for example, CGI (Computer Graphic Interface) commands. These may be executed by the drawing controllers 2e in the graphic display units each according to the FIFO buffer 2b of the common memory 2a

asynchronously with the processor 2c and drawing controller 2e. The host processor also actuates the PUSH pointer P1 of the FIFO buffer.

The FIFO buffer 2b has enough capacity to write the command or data group required for displaying a sheet of average graphic panel size and the host processor 1 terminates the drawing process without waiting for the end of drawing by the relevant graphic display unit and shifts sequentially to the process of the next graphic display unit.

In each graphic display unit $2_1 \sim 2_n$, the internal processor 2c compares the PUSH pointer P1 and POP pointer P2 locations of the FIFO buffer 2b within the common memory and registers the arrival of commands from the host processor 1.

The processor 2c reads high level commands written in the FIFO buffer 2b and executes processes such as keyboard communication. Moreover, the processor 2c also shares a part of the processes such as interpretation of high level commands, processing the commands which may be executed by the drawing controller 2e and control of drawing controller 2e, which would otherwise be done by the host processor 1, in order to alleviate burden of the host processor 1.

The drawing controller 2e draws command train on the frame buffer 2d in accordance with the commands given from the internal processor 2c and also reads such command train.

(b)~(e) show the timing for the internal processor 2c to receive command processing and execute the drawing in each graphic display unit $2_1 \sim 2_4$. The respective graphic display units execute in parallel command processings and drawing tasks, although the overhead for dealing with commands interpretation of commands and execution control is added.

(f) indicates, for reference, the time chart of the prior art in which the host processor 1 executes the command processing and drawing of the graphic display

units.

In comparison of (a) and (f), the reduction of the length of the time chart means alleviation of load shared on the host processor 1, resulting in a time surplus.

Fig. 5 is a flowchart indicating operations conducted by the driver of the host processor 1 through the common memory 2a, described above and provides an outline of the drawing task executed by the graphic display unit 2.

The driver of host processor 1 first outputs a command train to the FIFO buffer 2b of the common memory 2a for executing the write operation while renewing the PUSH pointer P1 command by command from the area indicated by PUSH pointer P1, until the end of write operation thereto (steps 1, 2). Upon completion of write operation, the notified interruption is issued to the drawing controller 2e (step 3). In the step 1, when the address reaches the end address during the write operation of one command, the write operation returns to the start and is continued. Such write operation in this step is carried out so that the PUSH pointer P1 does not follow-up the POP pointer P2. Moreover, in the step 2, renewal of PUSH pointer P1 is carried out in such a manner that the pointer P1 indicates the location next to the one command written in the FIFO buffer 2b.

The internal processor 2c and drawing controller 2e of the graphic display unit 2 decides whether the locations indicated by the PUSH/POP pointers P1, P2 are the same or not. When these locations are different, the processor 2c and controller 2e reach one command from the location of FIFO buffer indicated by the POP pointer P2 and interpret such command and then execute the drawing process.

Consequently, the location indicated by the POP point P2 is renewed to show the heading area of the command next to that read precedingly. When the PUSH/POP pointers P1, P2 indicate the same location, the read operation of command terminates.

During such a series of operations, the write operation to the FIFO buffer 2b and the read operation therefrom are carried out asynchronously and commands are stored until the FIFO buffer becomes full.

In this case, the control indicated below is carried out. Namely, the FIFO buffer previously has space area for one command and when the FIFO buffer becomes full, the driver stores therein the command to "notify the interruption of host processor" and at once terminates the output of the command train.

The graphic display unit sequentially reads contents of the FIFO buffer, carries out the drawing control and finally issues an interruption to the host processor in order to inform the drive of the existence of space in the FIFO buffer. Upon reception of this information, the driver starts again the write operation of the command train which has been suspended.

Fig. 6 is a conceptional view of a configuration of the principal portion of another embodiment of the present invention. This embodiment indicates the configuration that a plurality of FIFO buffers (two buffers, in this embodiment) 2b1, 2b2, respectively storing command trains in accordance with the processing priority level, are provided within the common memory 2a.

In such configuration, the driver in the host processor 1 writes, for example, the command train CA having a high priority level such as display of alarm information to the preference FIFO buffer 2b1, while it writes the command train CB for ordinary graphic display usually to the FIFO buffer 2b2.

The graphic display unit 2 decides first whether the command train is stored in the preference FIFO buffer 2b1 or not. In case it is stored herein, the unit 2 reads first the command train and executes the drawing process. On the other hand, when space is generated in the preference FIFO buffer, the graphic display unit 2 usually reads the command train stored in the FIFO buffer 2b2 and then executes the drawing process.

Fig. 7 is a flowchart indicating an example of the operations to be carried out by the graphic display unit in the embodiment of Fig. 6. Decision whether the command train is written in the preference FIFO buffer 2b1 or not is usually carried out for each execution of one command during the processing of the FIFO buffer 2b2.

With such configuration, the display having a higher priority level such as alarm information realises the drawing process with preference to the other graphic display.

According to such embodiment, the respective graphic display units share in parallel, in the form of distributed processing, a part of the processings to be intrinsically done by the host processor. Thereby, a burden of the host processor may be alleviated and the processing capability of the host processor can be improved as a whole.

Moreover, a configuration linking the host processor and graphic display units through the FIFO buffer provided within the common memory has eliminated the necessity for the graphic display unit to become the bus master realising simplification in configuration and inhibits the host processor to take the bus right, ruling out such a discrepancy that the processing speed of the graphic display unit is lowered.

Moreover, since the FIFO buffer is formed in the common memory in the form of software, this common memory can be used freely as a memory and a plurality of FIFO buffers may also be formed as required and the command train having the higher priority level may be processed in preference to the other command train.

Fig. 8 is a block diagram indicating an example of another configuration of the graphic display unit connected to the host processor through the common bus.

The graphic display unit of the prior art usually provides a ROM therein to store operation programs of the microprocessor. In this case, when it becomes apparent that the programs stored in the ROM includes bugs, it is

essential to conduct replacement of the ROM. Indirect bug elimination by programs from the host processor and replacement of an operation program including the loading of corrected programs from the host processor, results in a lowering of maintenance ability.

The embodiment of the present invention improves such problems. Namely, the ROM for storing operation (unit) programs to be executed by the internal processor is omitted for space saving and maintenance ability is as much improved.

In Fig. 8, the graphic display unit 2i has a common memory 2a, which is also used in common from the host processor 1, allowing the one to be connected with the common (system) bus BS and the other to be connected with the internal bus NBS. This common memory is so configurated that changing address space observed from the internal processor 2c can be assigned to two kinds of space. 2j designates external reset means which sets or cancels the reset mode of the apparatus with a command from the host processor 1; 2k, area select means for switching assignment of two kinds of spaces of the common memory 2a in accordance with an instruction from the host processor 1 and the internal processor 2c; and 2h, local memory connected with the internal bus NBS.

The common memory 2a is capable of dynamically changing assignment to two kinds of spaces of the reset vector area and ordinary memory area depending on the instruction of the host processor 1 or internal processor 2c. In the reset mode, the host processor 1 assigns the common memory 2a to the reset vector area, loads thereto the operation programs. After cancelling the reset mode, the internal processor 2c transfers the operation program loaded to the vector area to the local memory 2h, assigns the common memory 2a the ordinary memory area for transfer of high level command/data sent from the host processor 1.

Fig. 9 is a block diagram of configuration indicating an example of the graphic display unit applied

to the apparatus of Fig. 8.

The internal bus NBS is connected, in addition to the common memory 2a, internal processor 2c, local memory 2h, with drawing controller 2e and frame buffer 2d, colour look-up table 2g connected through such drawing controller 2e. Numeral 3 designates CRT as a display means.

The common memory 2a is so configurated that the address space thereof can be dynamically assigned to two kinds of space of the reset vector area and ordinary memory area, depending on the switching by a switch SW. Namely, the switch SW is enabled to any position of the positions a, b with an output of the area select means 2k. It is informed to the common memory 2a through the switch SW whether the address corresponds to that decoded by the address decoding means 2m and designates the reset vector area sent from the internal processor 2c or to that designated the ordinary memory area, so that the address space can be dynamically assigned to either space of the reset vector and ordinary memory areas.

With such configuration, the common memory 2a is placed to a certain fixed address as shown in Fig. 10 in the address space in the graphic display unit side observed from the host processor, while in the address space of memory observed from the internal processor 2c of the graphic display unit, the common memory is assigned to the reset vector area ($0) as shown in Fig. 11(a) when the switch SW is set to the position a by the area select means 2k or assigned to the ordinary memory area as shown in Fig. 11(b) when the switch SW is set to the position b.

Here, the local memory is also assigned to include the reset vector area, but if the switch SW is set to the position a, the overlap area is masked.

Operations of the apparatus having such configuration will be explained hereunder.

Fig. 12 is a flowchart indicating an example of the operations conducted by the host processor 1 and internal

processor 2c.

First, the host processor 1 sends a command to the external reset means 2j to reset the internal processor 2c in the graphc display unit 2i. Simultaneously, the host processor 1 sends a command to the area select means 2k, sets the switch SW to the position a and maps the common memory 2a to the reset vector area (step 1).

Next, host processor 1 loads a load program to the common memory 2a (step 2).

The host processor 1 sends a command to the external reset means 2j to start resetting of the internal processor 2c (step 3). Thereby, the program loaded to the common memory 2a is enabled to run.

Consequently, the internal processor 2c copies the program itself to the local memory 2h by the program and causes the program for loading the program itself to run after jumping to the copied program (step 4). The internal processor 2c first sends a command to the area select means 2k in accordance with a loading program of the program itself, sets the switch SW to the position b, remaps the common memory 2a to the ordinary memory area, uses this memory area as the buffer and loads the program itself to the local memory 2h from the host processor 1 (step 5).

In this case, the host processor 1 loads the program itself to the local memory 2h, in co-operation with the program loaded in the step 5, using the common memory 2a.

Consequently, the internal processor 2c starts operations as the graphic display unit in accordance with the program loaded to the local memory 2h (step 6).

This operation may be started by jump from the program loaded in the step 5 but in general by the reset signal from the external reset means 2j with instruction from the host processor 1 also for initialization of hardware.

In the consequent operations, the internal processor 2c uses the common memory 2a for receiving and

sending the high level command/data from/to the host processor 1 and shares a part of operations of host processor 1 by the high level command interface (for example, CGI) for graphic use.

In the above embodiment, the common memory 2a is connected at the one end directly to the system bus BS. For instance, however, such direct connection to the system bus is no longer necessary if designated address and data can be written through communication from the host processor 1.

As described, according to this embodiment, the ROM for storing operation program to be executed by the internal processor 2c and data may be eliminated from the graphic display unit 2i. Thereby, space saving can be realised and replacement of ROM is now unnecessary, improving the maintenance ability. Moreover, even when large capacity operation (unit) program is used, it may be booted up by preparing the capacity enough for loading the minimum boot program.

Since the operation program is executed after it is transferred to the local memory ensuring fast access speed, speed-down by competition of bus access can be avoided and performance can also be improved in comparison with the case where operation program is stored in the common memory and then it is executed.

Fig. 13 is a block diagram indicating the principal portion of another configuration example of the graphic display unit 2i. In the case of this example, a plurality of drawing controllers are used.

The graphic display unit which has been proposed provides a plurality of drawing controllers, processes in parallel the shared drawing commands to be processed by such drawing controllers, or processes in parallel the shared memory bits forming a pixel (for example, 8 bits/pixel). In such a graphic display unit, the end of command train is detected by polling the status of respective drawing controllers or by receiving the drawing command end interrupt from respective drawing

controllers.

The parallel processing type graphic display unit generates the same image with a plurality of drawing controllers or displays superimposed images by generating different images with respective drawing controllers. In case a plurality of drawing controllers generate the same image, the drawing command processings by a plurality of controllers must be synchronised in order to prevent disturbance of display. On the other hand, when respective drawing controllers generate respective images and superimpose them for display, such synchronisation is unnecessary.

The apparatus of the prior art has conducted such control only with the software processing, resulting in a problem of coexistence of acquisition of performance and synchronous drawing.

This embodiment has solved such problem by attaining; (a) synchronous drawing in such a case that a plurality of drawing controllers are used for creation of a drawing, (b) improvement in drawing performance in such a case that independent images are drawn by using a plurality of drawing controllers and these are displayed as a synthetic image.

In Fig. 13, 2e1, 2e2 designate drawing controllers having the function to interpret and execute the drawing command train given through common bus BS from the higher host processor 1 and also executes drawing on the frame buffers 2d1, 2d2. In this case, two drawing controller units are used.

2g designates a colour look-up table which is responsive to the input from frame buffers 2d1, 2d2; 2c, an internal processor as a display engine having the function to read image data stored in the frame buffers 2d1, 2d2 and guides such data to the colour look-up table 2g. The internal processor 2c is so configurated that two kinds of window type display, for example, the synthetic display of two images of 8 bits/pixel and 4 bits/pixel can be realised by changing on the realtime

basis the mapping of the colour look-up table 2g.

3 designates a display means such as CRT which displays the display data from the colour look-up table 2g.

71 designates an AND circuit which is responsive to the inputs notifying the end of drawing from a plurality of drawing controllers 2e1, 2e2; 72, an OR circuit which is responsive to the inputs notifying the end of drawing from a plurality of drawing controllers 2e1, 2e2; 8, an interrupt mode changeover switch which selects the signal from the AND circuit 71 and the signal from the OR circuit 72 and gives the selected signal to the higher host processor 1 as the interrupt signal. This changeover switch is formed by the software.

Here, for example, the changeover switch 8 is set to the side of contact a and the signal from the AND circuit 71 is selected for drawing on the window of two images of 4 bits/pixel with the drawing controllers 2d1, 2d2, meanwhile the changeover switch 8 is set to the side of contact b and the signal from the OR circuit 72 is selected for drawing on the window of 8 bits/pixel (respective drawing controllers 2e1, 2e2 share 4 bits and draw the same image simultaneously).

Operations of apparatus in such configuration are explained hereunder.

The drawing of 8 bits/pixel image will be explained. In this case, the changeover switch 8 is set to the contact b with a command from the host processor 1. Moreover, the drawing controllers 2e1, 2e2 receive almost the same drawing command train, except for the colour data, from the host processor 1 and processes such command train. If the processing of drawing command train is asynchronised and the processing of the drawing controller 2e1, for example, is delayed, code not defined appears in the synthetic image, flickering or disturbing the displayed image. In the apparatus of the present embodiment, the end of drawing message at the drawing controllers 2e1, 2e2 is applied as the interruption to

the host processor 1 through the OR circuit 72 and changeover switch 8. On the other hand, the host processor 1 having received such interruption outputs a command for queuing of events, causing two drawing controllers to operate synchronously.

Next, the changeover switch 8 is set to the contact a with a command from the host processor 1 for drawing of image on the window for synthetic two images of 4 bits/pixel. In this case, the drawing controllers 2e1, 2e2 may be operated independently and the end of drawing message is also independently applied to the host processor 1 through the AND circuit 71 and changeover switch 8.

As described, the host processor 1 connects the changeover switch 8 to the contact b for drawing of image of 8 bits/pixel to give the signal from the OR circuit 72 by interruption, then applies the drawing command train to the drawing controllers 2e1, 2e2 and then connects the changeover switch 8 to the contact a upon reception of the drawing end interrupt signal. Thereby, deterioration of performance can be prevented by eliminating disturbance of display due to defective synchronisation of drawings.

Fig. 14 is a block diagram of another example of a configuration of graphic display unit.

This embodiment realises synthetic display of the standard bit map window and a special window for emulating the screen consisting of character display and graphic display.

The existing graphic display unit of this type is formed with the character plane and graphic plane. Such existing apparatus is capable of independently controlling the character display and graphic plane and therefore it provides such a characteristic that the reversible edition can be done by scrolling only the character display with the graphic display fixed.

Since the character display generally allows rewriting at a high speed, when a table which changes

from time to time is to be displayed, such application has been made so that only the background is displayed with the graphic plane and character data to be altered is often displayed by the character plane.

However, provision of two kinds of planes of character plane and graphic plane results in a problem that capacity of hardware is as much increased. Moreover, since the LSI which assures high speed drawing and display of characters is recently available even in the graphic plane, it is now required to configurate the graphic display unit only with the graphic plane.

In this case, however, if for the one pixel, several bits are used uniquely, it is impossible to scroll only the character in case the character and graphic are displayed simultaneously, resulting in a problem that perfect emulation cannot be executed.

This embodiment has solved such problem and enables that the graphic display and character display are superposed on one window of a multi-window display. Such displays can be controlled individually and this window and other standard windows can be displayed simultaneously.

In the example shown in Fig. 14, a number of bits of one pixel is selected to 8 bits.

Two frame buffers 2d1, 2d2 respectively employ four bits for one pixel. For example, the DRAM of 2 MB is used.

A pair of drawing controllers 2e1, 2e2 for drawing/display control coupled with frame buffers 2d1, 2d2 include therein a window attribute output means 20. This makes drawing on the corresponding frame buffer, during the drawing mode, dependent on the command from the internal processor 5 functioning as the command output means. The multi-window on the display screen is also formed, during the display mode, by programmably reading data from the frame buffer.

As such a drawing/display co-processor, the CPU-82786 by Intel, for example, which outputs the window

status signals of one bit indicating the window attribute from the terminal W and also outputs the video data of 4 bits/pixel from the data output terminal DO, can be used.

The colour look-up table 2g is composed of entries 0~511. Of these entries, the half 0 255 is used for storing colour data to display the ordinary window of 8 bits/pixel and the remaining half 256~511 is used for superimposed display of two images of 4 bits/pixel.

Moreover, the window status signal from the drawing controller 2e1 for controlling drawing/display is assigned to the most significant bit (MSB) of the colour look-up table 2g to change the map in accordance with the attribute information of the window.

Here, the colour look-up table 2g prepared 8 bits for each colour of red R, green G and blue B and the colour data read from each table is output to the CRT 3 as the display means through the D/A converters 31, 32, 33.

2a designates a common memory used in common from the host processor 1 and internal processor 2c, to which the parameters for drawing are loaded from the host processor 1.

Operations of the apparatus thus configurated are explained next hereunder.

In the following explanation, it is assumed that the terminal emulation window of the character plane and graphic plane is operated.

As shown in Fig. 15, the standard window is used when the window attribute signal (window status signal) from the window attribute output means 20 is "0" and the terminal emulation window is used when such signal is "1". The colour look-up table 2g becomes the area for storing colour data for the standard window in the entries from 0 to 255, or becomes the area for terminal emulation in the entries from 256 to 512.

While the drawing controllers 2e1, 2e2 are scanning the standard window, the data of 8 bits/pixel refers to the table for the standard window formed in the area of

entries 0 255, and reads the signal intensity data of R, G, B written therein. This data is D/A converted to form the colour signals and are finally displayed on the CRT 3.

The window for terminal emulation realises the character plane with the one drawing controller 2e1 of the two drawing controllers 2e1, 2e2 and the graphic plane with the other drawing controller 2e2, respectively on the basis of 4 bits/pixel.

Here, in the case of the superposed display of the character plane and graphic plane, the character display area is given the priority to the graphic plane and character display is carried out on the graphic plane. In case no-display of character plane is indicated by the code "0, 0, 0, 0", the colour look-up table 2g is enough when the colour code of graphic plane is defined only in the area of "1, 0, 0, 0, 0, X, X, X, X" (here, the header 1 represents the terminal emulation area and X represents 1 or 0).

Fig. 16 is an example of the video data given to the colour look-up table 2g, in which "X,X,X,X" of the 0th~3rd bits represents video data (4 bits/pixel) given from the drawing controller 2e1 for the graphic plane, while "0,0,0,0" of 4th~7th bits, the video data (in this case, the code indicating that character is not displayed) given from the drawing controller 2e1 for the character plane. The 8th bit (MSB) is given the signal representing the window attribute drawn on the frame buffer (in this case, "1" representing the terminal emulation window) from the window attribute output means 20, while the display scanning operation of the window is carried out.

When the video data indicated here and the signal representing window attribute are given to the colour look-up table 2g, only the data of the graphic loop-up plane is displayed.

Fig. 17 is a conceptional view of the superimposed display of the character plane and graphic plane.

(a) is the character plane drawn on the frame buffer 2d1 by the drawing controller 2e1, and the video data output, during display scanning, from the drawing controller 2e1 is output in such a manner that the colour code other than "0,0,0,0" is output for the character part, while the colour code "0,0,0,0" is output for the part other than the character part.

(b) is the graphic plane drawn on the frame buffer 2d2 by the drawing controller 2e2.

The drawing controllers 2e1, 2e2 respectively receive commands from the internal processor 2c and carry out the drawing on the character plane and graphic plane and display scanning in accordance with such commands.

As a result of superposed display of the plane of (a) and the plane of (b), the character and graphic are mixed as shown in (c) and when these are superposed, the character is preferentially displayed.

Moreover, character plane and graphic plane are independently controlled by the drawing controllers 2e1, 2e2 allowing individual scrollings.

Fig. 18 is an example of definition written in the terminal emulation area (up to the entries 256~512) of the colour look-up table 2g.

In the area defined by $101 to $10F, the graphic plane display colour is defined so that only the graphic is displayed without character display. In addition, in the area defined by $110~$1FF, the character plane colour is defined.

In the area of $100, black (colourless) is defined and both character plane and graphic plane are colourless.

Here, the display priority of divided pairs of n bits/pixel can be changed by updating contents of colour look-up table 2g. Therefore, it is possible to momentarily change the display in such a way that the graphic is preferentially displayed on the character in place of the display of character with priority to the graphic display.

In the above explanation, the terminal emulation window of the character plane and graphic plane is operated, but in case all bits of one pixel are drawn by the drawing controllers 2e1, 2e2 like the ordinary bit map window, the same command is given simultaneously to all drawing controllers from the internal processor to conduct the parallel processing in order to realise high speed drawing/display.

In the above embodiment, the status signal of one bit is output from the window attribute output means as the window attribute information but each window may be given individual colour maps by providing a variety of signals which can be recognised for each window as the window attribute information.

Moreover, a pair of drawing controllers are used in the above embodiment, but much more drawing/display co-processors may also be used.

According to this embodiment, the windows superposing a plurality of planes such as bit map window and terminal emulation window, etc. may simultaneously be displayed.

CLAIMS:

1. A graphic display unit, in a system connecting a plurality of graphic display units to a host processor via a common bus, comprising a 2-port common memory connected to the common bus at the one terminal and to an internal bus at the other terminal, a part of this common memory being in the form of a FIFO (First-In First-Out) buffer for transmission of high level command/data to the host processor.

2. A graphic display unit according to Claim 1, in association with a plurality of similar units connected to the common bus wherein the FIFO buffers in the several units provide a plurality of levels in accordance with a required priority of processing.

3. A graphic display unit according to Claim 1, wherein the other terminal of the common memory is used for storing program codes or transmission of information.

4. A graphic display unit according to Claim 1, comprising: an internal processor coupled to the internal bus and functioning as the command output means; frame buffers, drawing controllers coupled with the internal bus, drawing a command train on the frame buffers in accordance with the commands given from the internal processor and reading such command train, and a colour look-up table which is responsive to video data output from the frame buffers and outputs the data therefrom to display means.

5. A graphic display unit having an internal processor to display graphic on a display means by receiving drawing command/data sent from a host processor, comprising a common memory arranged to be used in common from the host processor and is so configurated that dynamic assignment of addresses observed from the internal processor to two kinds of set vector area and ordinary memory area can be changed, an external reset means for setting or cancelling a reset mode of the apparatus in accordance with the commands from the host processor, an area select means for switching the

assignment of two kinds of spaces of the common memory in accordance with the commands from the host processor and internal processor, and a local memory connected to the internal processor and common memory through an internal bus, wherein the host processor assigns the common memory to the set vector area under the reset condition of the external reset means and loads thereto the operation program, and the internal processor transfers under the cancelling of the reset mode by the external reset means, the operation program loaded to the vector area of the common memory to the local memory, assigns the common memory to the ordinary memory area and also uses the common memory for transmission of high level command/data sent from the host processor.

6. A graphic display unit which comprises a plurality of drawing controllers, each of which interprets and executes the drawing command train given from a host processor and has a function to draw an image on an image memory, further comprisng AND circuits which are responsive to a drawing end message sent from a plurality of drawing controllers; OR circuits which are responsive to a drawing end message sent from a plurality of drawing controllers; and an interrupt mode changeover switch for selecting a signal from an AND circuit and a signal from an OR circuit in accordance with a method of image creation with the signal from the host processor and giving the selected signal to the host processor as the interrupt signal.

7. A graphic display unit comprising frame buffers formed by a plurality of bits per pixel; a plurality of drawing controllers for dividing a plurality of bits and independently executing the drawing to respective pairs; a window attribute output means for outputting window attribute information drawn in the frame buffers while the display scanning opeations of the relevant window are carried out; a colour look-up table arranged to respond to the attribute information sent from the window attribute output means and the video data read from the

frame buffer, is arranged to extract colour codes of respective pairs in compliance with a dividing method with the window attribute information and then to display such colour codes on the display means; and an internal processor arranged individually to give commands to a plurality of drawing controllers and to give the same command at a time to all drawing controllers.

8. A graphc dislay unit according to Claim 7, wherein the internal processor is coupled with a host processor through the common memory to be used in common also by the host processor.

9. A graphic display unit substantially as hereinbefore described with reference to Figs. 3 to 5 of the accompanying drawings.

10. A graphic display unit substantially as hereinbefore described with reference to Figs. 6 and 7 of the accompanying drawings.

11. A graphic display unit substantially as hereinbefore described with reference to Figs. 8 to 12 of the accompanying drawings.

12. A graphic display unit substantially as hereinbefore described with reference to Fig. 13 of the accompanying drawings.

13. A graphic display unit substantially as hereinbefore described with reference to Figs. 14 to 18 of the accompanying drawings.